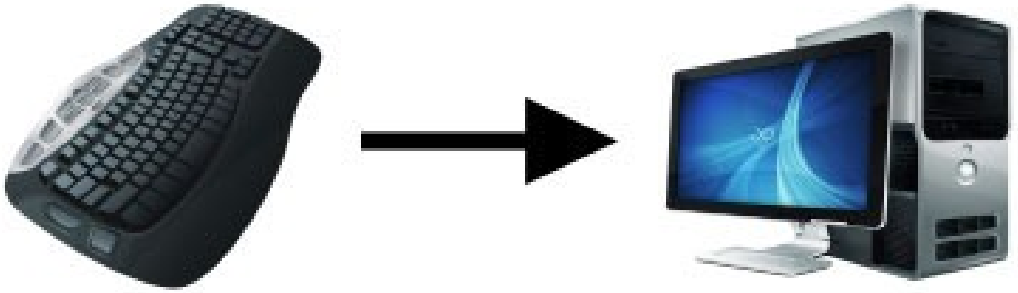
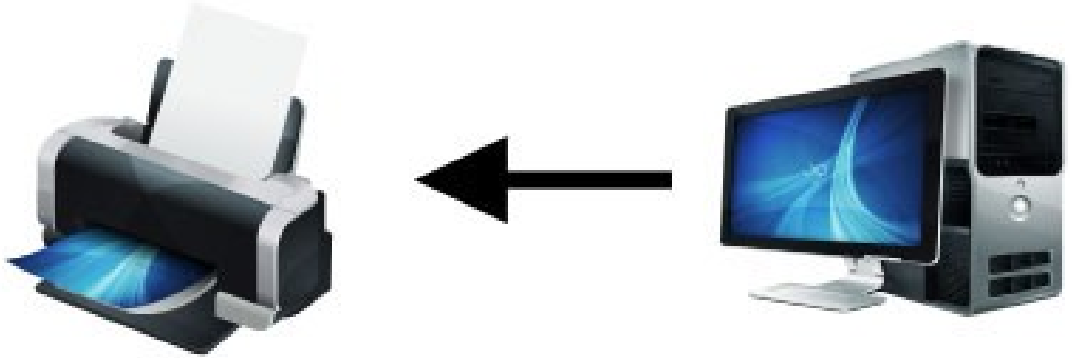
Ввод и вывод данных

Пример ввода



Пример вывода



Компьютерная программа

Это список команд (инструкций) для компьютера. Команды могут быть любыми, например:

- считать информацию с клавиатуры;
- произвести арифметические вычисления (+, –, *, /);
- вывести информацию на экран.

Например, фрагмент программы на Паскале:

```
 readln(a);        // чтение (ввод) данных
readln(i,j,k); // чтение (ввод) данных
c:=a*(i+j-k); // вычисление значения выражения
writeln(c);  // вывод данных на экран монитора
```

Можно ли обойтись без ввода данных?

- **Допустим, a=10, i=6,j=3, к=4**
- **В программе: `c:=a*(i+j-k);`**

c=10(6+3-4)=10*5=50*

- **Но программа должна работать для любых значений a,i,j и k.**

Ввод данных с клавиатуры

Read(a) — вводит с клавиатуры в программу значения переменной **a** в соответствии с **ее типом** (если это целое число, то, например -10; если вещественное, то – 10.6)

Readln(i,j,k) — вводит с клавиатуры в программу значения **трех** переменных в соответствии с их типом с новой строки.

При выполнении операторов **READ** или **READLN** компьютер переходит в режим ожидания данных:

нужно **ввести данные с клавиатуры и нажать клавишу ENTER.**

Операторы read и readln

Например, в программе переменные объявлены:

```
Var a:integer; b,c:real;
```

Оператор `read(a,b,c)` означает, что с клавиатуры нужно ввести числа через пробел и нажать клавишу **ENTER:**

```
10  12.5  -3.7
```

В программе эти переменные будут иметь значения:

a=10 b=12.5 c=-3.7

Отличие оператора **read** от **readln** состоит в том, что после выполнения **readln** переходит **на новую строку для ввода данных.**

Вывод данных

Без оператора вывода данных мы не увидим результата выполнения программы.

`Write(c)` — **выводит** на экран монитора значение переменной с.

`Writeln(a,b)` — **выводит** на экран монитора значение двух переменных с новой строки.

Операторы Write и Writeln

- `write(a,b)` – выводит значения в одной строке
- `writeln(a);writeln(b)` – каждое значение выводится с новой строки.

Пример: пусть в программе `c:=50`
Оператор `writeln(c)` выведет значение на экран:
50
Оператор `writeln('c=',c)` выведет значение на экран:
c=50

Примеры вывода данных

- Если необходимо вывести числа через пробел:

```
writeln(a,' ',i,' ',j,' ',k)
```

- Пусть в программе a=10 c=50, необходимо вывести на экран: **a=10 c=50**

```
writeln('a=',a, 'c=',c)
```

- Если нужно на вывод каждого числа выделить 5 позиций:

```
writeln(a:5,c:5)
```

Примеры вывода данных

Если нужно вывести текст, то текст заключается в апострофы:

- `Writeln(' Введите значение a');`

На экране появится: **Введите значение a**

В программе должен быть оператор ввода:

read(a);

Далее – нужно ввести значение a, например 10 и нажать клавишу **ENTER**

Пример программы

```
Var a,c:real; i,j,k:integer;  // блок описания

 begin                              // начало программы
 writeln('введите значение a');
 readln(a);                         // чтение данных
 writeln('введите значения i,j и k');
 readln(i,j,k);                     // чтение данных
 c:=a*(i+j-k);  // вычисление значения выражения
 writeln('c=',c);    // вывод значения на экран
 end.              // конец программы
```

Символьные переменные в Паскале

Вычислительные машины имеют дело не только с числами, но и с **текстом.**

Для обработки **отдельных символов текста (букв)** используется тип данных **CHAR**(символ) .

Для обработки строки текста(**предложения)** используется тип данных **STRING** (строки).

В программе символы и строки должны быть описаны:

Var stroka:string; c:char;

Строки можно вводить и выводить, сравнивать, соединять.

Пример записи присваивания символьных значений

```
st:='Привет!';
sta:=' Как тебя зовут?';
sta1:=st+sta;
```

В результате выполнения фрагмента программы переменная sta1 будет иметь значение 'Привет! Как тебя зовут?'. Знак «+» в данном случае для символьных переменных работает как команда сцепления строк.

Операция конкатенации (+)

Для символьных констант и переменных определена операция — **сложение** или **конкатенация строк**. Она обозначается знаком + и позволяет строить из нескольких слов или строк одну — результирующую.

Например, в программе:

```
a:=' береги ';
b:=' честь ';
c:= ' смолоду';
d:=a+b+c;
writeln(d);
```

На экране будет: береги честь смолоду

Пример программы с использованием ввода символьных переменных

```
Program text2;
Var at, bt, ft:string;
begin
Writeln(' Введите ваше имя');
Readln(at);
Writeln(' Введите вашу  фамилию');
Readln(bt);
Writeln('Здравствуйте, ',at,'  ',bt,
',’Начнем урок!');
End.
```

Пример программы с использованием символьных переменных

```
Var at, bt, ft:string;
begin
at:=' Быть или не быть - ';
bt:=' Вот в чем вопрос!';
ft:= ' У.Шекспир';
Writeln(at,bt);
Writeln (ft);
end.
```

Задание 1. Составьте программу для создания следующего диалога:

Введите ваше имя **Сергей**

Введите ваш возраст **14**

На экран нужно вывести:

Сергей, 14 лет — это хороший возраст.

*Данные, выделенные **красным цветом** должны быть введены с клавиатуры.*

Задание 2. Составьте программу для создания следующего диалога:

Добрый день!

Назовите, пожалуйста, свое имя

Саша

Рад познакомиться, Саша!

Как ты поживаешь?

Хорошо

Какое совпадение, и я тоже хорошо.

Данные, выделенные красным цветом должны быть введены с клавиатуры.

Задание 3. Написать программу, которая выводит на экран следующий текст:

Как тебя зовут? *{Ввести имя}*
В каком классе ты учишься? {Ввести класс}

Привет **ИМЯ !**
Ты учишься в **№** классе и мы будем изучать Паскаль.

Задание 4.

Написать свою программу-диалог с компьютером, которая содержит **не менее 5 вопросов.**
В диалоге желательно использовать конкатенацию строк.